Generation of reduced words of permutations

Olcay Coşkun and Müge Taşkın
olcay.coskun@boun.edu.tr and muge.taskin@boun.edu.tr

Boğaziçi Üniversitesi Matematik Bölümü Bebek, Istanbul 34342 Turkey

Abstract

We introduce a new algorithm to produce all reduced words of a given permutation by putting an order on the consecutive applications of the braid relations. Thus, starting from a canonical word associated to a permutation via its tower diagram, we obtain all reduced expressions, in a tractable and explicit way.

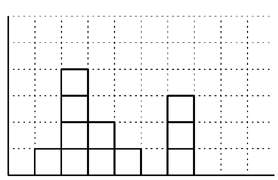
1. Introduction

The symmetric group S_n on $[n] = \{1, 2, \ldots, n\}$ is generated by adjacent transpositions $\{s_i : i = 1, 2, \ldots n-1\}$, where s_i stands for the transposition $(i, i+1)$. Given any permutation $\omega \in S_n$, an expression $s_{i_1} s_{i_2} \ldots s_{i_l}$ representing ω is called a word for ω. Such an expression with minimal k is called a reduced word for ω. The index k is determined by ω and any two reduced words for ω are related by the braid relations given as follows.

1. (Short braid relation) $s_i s_j = s_j s_i$ for any i, j with $\mid i - j \mid > 2$.
2. (Long braid relation) $s_i s_{i+1} s_i = s_{i+1} s_i s_{i+1}$.

Throughout the paper, we abbreviate this expression as $i_1 i_2 \ldots i_k$.

In [2], the authors introduced tower diagrams as a new combinatorial object to study reduced words of finite permutations. A tower diagram is a diagram that grows out of the positive x-axis in the plane. For example,



is a tower diagram. One of the main results on tower diagrams is the existence of an explicit bijective correspondence between tower diagrams and permutations. The mutual inverse bijections are given by two algorithms. The sliding algorithm allows one to slide all reduced words of a given permutation into the empty

[1]Both of the authors are supported by TUBITAK-109T661 and Boğaziçi-BAP-6029.

diagram to obtain certain labelings of a unique tower diagram, called *standard tower tableaux*, and the flight algorithm allows one to read all standard tower tableaux of the given tower diagram, to obtain all reduced words of a unique permutation. Thus we also obtain the bijection between tower diagrams and the permutations.

Using a canonical standard tower tableau of a given tower diagram, we obtain a unique reduced word, called *the natural word*, which can be characterized as a sequence of increasing sequences of consecutive integers with decreasing first terms. See Section 2 for further details.

The aim of this paper is to introduce an algorithm to describe all reduced words of a given permutation. The main tool for the algorithm is the natural word of the given permutation. The need for such an algorithm comes from the chaotic behavior of the application of the braid relations. Recall that, by definition, any two reduced words of a permutation are braid related, which means that one can start with a reduced word and apply a series of braid relations to obtain all reduced words. However this procedure of applying a sequence of braid relations is non-trivial and is not recursive, actually it is chaotic. Moreover the only stopping criterion is the determination of the number of reduced words, for which there is no simple formula. These make it difficult to apply braid relations to obtain the complete list of reduced words.

In the literature, there are other combinatorial objects to determine the set of all reduced words of a given permutation, such as balanced labeling of the Rothe diagram [3] and RC graphs [1] of permutations and the plactification map [4]. On the other hand, in both, underlying process is recursive and has no simple formulation. Note that the similarities between the Rothe diagram and the tower diagrams is discussed in [2].

On the other hand, our algorithm starts with the natural word and follows explicit and tractable steps to produce all reduced words, hence puts an order on the chaos created by the braid relations. The algorithm originates from another characterization of the natural word, proved in Section 3. According to this characterization, the natural word is the unique maximal word in the ***directed-braid poset*** of all reduced words of a given permutation, where the underlying partial order is obtained by putting a direction on the applications of the short and long braid relations. Hence the natural word can be interpreted as the unique word which is free from directed-braid relations.

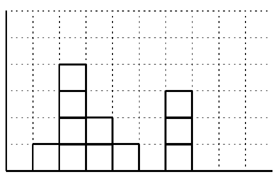
Via this characterization of the natural word and directed-braid poset, it is clear that there is at least one route from the natural word to any other reduced word. To obtain the generation algorithm for all reduced words, we exhibit a canonical such route. At the first step, we produce ***basic words*** of the given tower diagram. This step is basically the application of the long braid relation to the natural word. After determining the set of all basic words, it only remains to apply the short braid relation, which is done by the ***restricted shuffle*** on basic words. This step is a variation of the well-known shuffle operation on words.

The paper is organized as follows. In Section 2, we review tower diagrams and in Section 3, we introduce the directed-braid poset together with its basic

properties. The next section introduces the generation algorithm in two steps, as described above. The proof of the main theorem is also contained in this section. To illustrate our algorithm, we include two examples in the last section. One of the examples is the reduced words of the longest permutation in S_4.

2. Preliminaries on tower diagrams

In this section, we recall the necessary background from [2] without details. To begin with, a ***tower diagram*** is a finite sequence $\mathcal{T} = (\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_l)$ of towers where by a ***tower*** $\mathcal{T}_i$ of size $k_i \geq 0$ we mean a vertical strip of k_i squares of side length 1, for each $1 \leq k \leq l$. We always consider the tower diagram $\mathcal{T}$ as located on the first quadrant of the plane so that for each i, the tower $\mathcal{T}_i$ is located on the interval $[i-1, i]$ of the horizontal axis. It is also possible to represent a tower diagram $\mathcal{T} = (\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_l)$ with *weak decomposition* $(k_1, k_2, \ldots, k_l)$, obtained by listing corresponding tower sizes. For example, the tower diagram $(0, 1, 4, 2, 1, 0, 3)$ can be represented as follows:

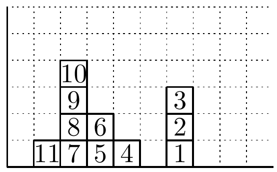


There are two basic operations on tower diagrams. We refer to [2] for details of these constructions. However note that one of them, the sliding algorithm, is a way to increase the number of cells in a tower diagram. By using this algorithm any reduced word of a given permutation w is slided to the empty diagram to obtain certain labellings of the unique tower diagram $\mathcal{T}_w$, that we call as *the standard tower tableaux* of shape $\mathcal{T}_w$. The other one, the flight algorithm, is a way to decrease the number of cells in a tower diagram $\mathcal{T}$ and when it is applied to the standard tower tableaux of shape $\mathcal{T}$, one can obtain the set of all reduced words of a unique permutation. With these correspondences, one also constructs a bijection between the set of all finite permutations and the set of all finite tower diagrams is obtained. We do not need the details of this construction in this paper, except the following special case.

Let $\mathcal{T}$ is a tower diagram of size n. Label the top cell of the left most non empty tower of $\mathcal{T}$ have label n and continue to label the cells from top to bottom and left to right, by decreasing the label by one at each step. Such labeling still gives a standard tableau of shape $\mathcal{T}$ which we call as **canonical tower tableau**. On the other hand the corresponding reduced word associated by the flight algorithm is called the **natural word** $\eta_{\mathcal{T}}$ of $\mathcal{T}$. In fact one can obtain the natural word by the following easy rule: Let η_i is the sum of the coordinates of the south-east corner of the cell with label i. Then

$$\eta_{\mathcal{T}} = \eta_1 \eta_2 \ldots \eta_n$$

For example, the canonical labeling of the above tower diagram is given as follows.



and the corresponding natural word is 78954534562.

With the above mentioned bijection, any permutation has a natural word. This word can also be characterized as the unique reduced word of the permutation which consists of increasing subsequences of consecutive integers with decreasing first letters, cf. [2, Proposition 5.1]. We will use this characterization in the rest of the paper.

Finally we introduce several notions regarding the reduced words. Let $\beta = b_1b_2\dots b_l$ be a word. We call β a **tower word** if for any index $i, 1 < i \leq l$, we have $b_i = b_{i-1}+1$, that is, if β is an increasing sequence of consecutive integers. It is clear, from the construction in the above discussion, that if β is a tower word, then it is the natural word of a tower diagram with a unique tower of positive length. For example, 4567 is a tower word.

It is also clear that any word $\beta = b_1b_2\dots b_l$ can be written uniquely as a product of tower words, say as $\beta = \beta_1\beta_2\dots\beta_s$ for some s, where β_i is called i-th tower word in β. We call this expression the **tower decomposition** of β. For example, if $\alpha = 4593465894$, then the tower decomposition is given by

$$\alpha = 45\ 9\ 34\ 6\ 5\ 89\ 4.$$

Evidently, the tower words in the decomposition of a word β corresponds to the towers of its tower diagram if and only if the word is a natural word.

3. The directed-braid poset of a permutation

In this section, we introduce the directed-braid poset $\mathcal{B}(\omega)$ of a permutation ω. To begin with, let $\mathrm{Red}(\omega)$ denote the set of all reduced words of ω. The two braid relations induces two different partial orders in this set as follows.

Let $\alpha = a_1a_2\dots a_k, \beta = b_1b_2\dots b_k \in \mathrm{Red}(\omega)$. Then we write $\alpha \leq_1 \beta$ if there is a unique index i such that

1. $a_{i+1} - a_i \geq 2$,
2. $a_j = b_j$ for any $j \neq i, i+1$, and
3. $a_i = b_{i+1}$ and $a_{i+1} = b_i$.

Clearly this covering relation refers to the short braid relation and also puts a restriction on the direction that we can apply it. To define the second relation, we first decompose the words into their tower words as $\alpha = \alpha_1\alpha_2\dots\alpha_s$ and $\beta = \beta_1\beta_2\dots\beta_t$. Then we write $\alpha \leq_2 \beta$ if either $s = t$ and there is a unique index i such that

1. $\alpha_j = \beta_j$ for any $j \neq i, i+1$, and
2. $\alpha_i = \beta_{i+1}$ and $\widetilde{\alpha}_{i+1} = \beta_i$

where $\widetilde{\alpha_i}$ is obtained by adding one to each letter of β_i, or $s = t+1$ and there is a unique index i such that

1. $\alpha_j = \beta_j$ for any $j < i$ or $j = i+1$,
2. $\alpha_j = \beta_{j+1}$ for any $j > i+1$, and
3. $\beta_i = \alpha_i \widetilde{\alpha}_{i+2}$.

Remark. It is almost as clear that the relation $\leq_2$ refers to the long braid relation. Indeed in the first case, where there are the same number of towers in the given words, a tower is moved to the right of the tower on its right with the rule that each of its letters is decreased by one. In the second case, there is a tower some part of which makes the above movement and hence the number of towers is increased by one. Clearly these are long braid moves. Note also that if the second case occurs, then also the first case would occur, that is, if some part of a tower can move to the right, then all of the tower can also move. See Section 4.1.1 for more explanation.

Now we write $\mathcal{B}(\omega)$ for the set $\mathrm{Red}(\omega)$ together with the reflexive closure of the relation $\leq_{d-braid}$ which is generated by the above covering relations. Therefore, we write

$$\alpha \leq_{d-braid} \beta$$

if either $\alpha = \beta$ or there is a sequence $a_0, a_1, \ldots, a_m$ of reduced words such that $a_0 = \alpha, a_m = \beta$ and for any $i, 0 \leq i \leq m_1$, we have either $a_i \leq_1 a_{i+1}$ or $a_i \leq_2 a_{i+1}$. We have the following result.

Lemma 3.1. *The above relation $\leq_{d-braid}$ on $\mathcal{B}(\omega)$ is a partial order.*

Proof. Reflexivity and transitivity of the relation follows directly from the definition. We only prove that the relation is anti-symmetric. It is easy to observe that if $\alpha \lneq_i \beta$, for $i = 1, 2$ then $\alpha \lneq_{lex} \beta$ under the lexicographic order. Therefore one can not have $\beta \leq_i \alpha$, for $i = 1, 2$ at the same time.

□

The main result of this section is that the poset $\mathcal{B}(\omega)$ has a unique maximal element. As we have described above, this result is the starting point of an algorithm to generate reduced words from the natural one.

Proposition 3.2. *The natural word η_ω of ω is the unique maximal element of the poset $\mathcal{B}(\omega)$.*

Proof. Let $\alpha = \alpha_1 \alpha_2 \ldots \alpha_k$ be a maximal element in $\mathcal{B}(\omega)$ given in its tower decomposition. Now since α is maximal, there is no word β such that $\alpha <_1 \beta$. But this is only possible if for any $i, 1 \leq i < k$, we have

$${}_l\alpha_i > {}_f\alpha_{i+1}$$

where we write ${}_f\alpha_{i+1}$ for the first letter of α_{i+1} and ${}_l\alpha_i$ for the last letter of α_i. Indeed, otherwise, ${}_l\alpha_i < {}_f\alpha_{i+1}$ and hence ${}_l\alpha_{i+1} - {}_f\alpha_i > 2$ since these are in different tower words. Therefore we can interchange ${}_l\alpha_i$ and ${}_f\alpha_{i+1}$ to obtain a greater word.

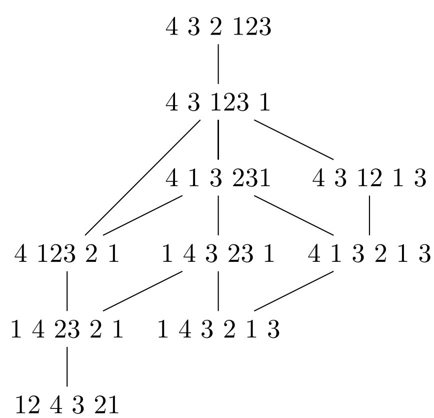
On the other hand, there is no word β such that $\alpha <_2 \beta$. But this is only possible if for any $i, 1 \leq i < k$, we have

$${}_f\alpha_i > {}_f\alpha_{i+1}.$$

Indeed, otherwise, ${}_f\alpha_i \leq {}_f\alpha_{i+1}$ and hence we can interchange α_i and α_{i+1} to obtain a greater word.

Thus we have proved that any maximal element α in $\mathcal{B}(\omega)$ has the property that in the tower decomposition of α the sequence of initial letters is decreasing. But by [2, Proposition 5.1], there is a unique word with this property, namely the natural word. $\square$

Although there is a unique maximal in the braid poset, there might be many minimal elements. An example of a braid poset with two minimal elements is the poset of the word 432123, where the minimals are the words 124321 and 143213. The full poset in this case is given as follows.



Remark 1. The above result tells us that it is possible to generate all reduced words starting from the natural word. The question is the existence of a canonical path from the maximal element to the chosen one. The advantage we have here is that via the directed-braid poset, we insist a direction on the braid relations: The short braid relation is the equality $s_i s_j = s_j s_i$ if $\mid j - i \mid \geq 2$, but in the braid poset, to go from the natural word to an arbitrary word, we can only interchange j and i if j is smaller than i. Similar comment is true for the long braid relation. Therefore, a maximal element in this poset is a word on which the directed-braid relations cannot be applied. In this sense, the unique maximal, the natural word, is the only braid-free word. Hence one would expect to have a canonical route from the natural word to any other reduced word and vice versa. For the rest of the paper, we explain such canonical routes. and vice versa

4. Conversion algorithm: From a reduced word to the natural word

By the Proposition 3.2, the natural word of a permutation w is the unique maximum among all reduced words of w in the directed-braid poset. In the following we will introduce two algorithms, by which one obtains from any reduced word α of w the natural word η_w as a chain this poset. The reason for this choice of algorithm will be clear in the following section.

4.1. An algorithm on $\leq_1$.

We will first apply $\leq_1$ on α to get a chain of words

$$\alpha = \alpha^0 \leq_1 \alpha^1 \leq_1 \ldots \alpha^s = \gamma$$

where the last one γ satisfies that when it's considered with its tower word decomposition, the first number in a tower is always less than the last number in the previous tower of γ, in fact their difference is always greater than or equal to 2. We will call such a word **a natural basic word** of w. In fact the natural word of a permutation w is a unique natural basic words satisfying that the first numbers in each of its tower words is strictly decreasing.

Assuming that first j words are already obtained in the above chain, we construct α^{j+1} as follows: Let a be the smallest integer that a tower word in

$$\alpha^j = a_1 \ldots a_n$$

can start with and let $1 \leq i_1, i_2, \ldots, i_k \leq n$ be the positions of a in α^j. We first start with the right most tower word starting with a, namely

$$a_{i_k} a_{i_k+1} \ldots a_{i_k+r} = a(a+1)\ldots(a+r)$$

for some $r \geq 0$, and apply one of the following passing rules in order to get a new word.

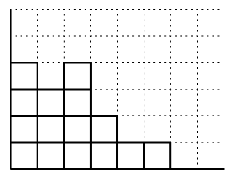
- If $a_{i_k} a_{i_k+1} \ldots a_{i_k+r}$ is the right most tower word of α^j or if the right next tower word to $a_{i_k} a_{i_k+1} \ldots a_{i_k+r}$ starts with a number less than $(a+r)$, namely

  $$a_{i_k+r+1} < a_{i_k+r}$$

  then we continue by applying the same algorithm with the next tower word starting with $a_{i_{k-1}} = a$.
- If $a_{i_k+r+1} \geq a_{i_k+r} + 2 = (a+r)+2$ then α^{j+1} is obtained by moving the tower word $a(a+1)\ldots(a+r)$ to the right of a_{i_k+r+1} in α^j.
- If none of the tower words starting with a in α^j can move to the right subject to the above rule, then the same algorithm is applied this time on the smallest integer bigger than a in α^j, that a tower word can start with.

One can easily observe that the algorithm terminates if and only if for any consecutive tower words in α^j, the first number in the latter is always smaller then the last number of the preceding one i.e., if the word is a natural basic word.

Example 4.1. Let $\alpha = 134521321654321$. One can easily show by using sliding algorithm that α is a reduced word of the permutation whose tower diagram is the following



where the corresponding natural word is $\eta = 6\ 5\ 45\ 3456\ 234\ 1234$, given with its tower decomposition. In the following we will apply $\leq_1$ to produce a natural basic word γ. Below the bold numbers in each step represent the towers words which are subject to move according to the above algorithm.

$$\begin{aligned}
\alpha = \alpha^0 &= 1\ \ 345\ \ 2\ \ 1\ \ 3\ \ 2\ \ \mathbf{1}\ \ 6\ \ 5\ \ 4\ \ 3\ \ 2\ \ 1 \\
\equiv \alpha^1 &= 1\ \ 345\ \ 2\ \ 1\ \ 3\ \ 2\ \ 6\ \ \mathbf{1}\ \ 5\ \ 4\ \ 3\ \ 2\ \ 1 \\
&\ \ \vdots \\
\equiv \alpha^4 &= 1\ \ 345\ \ 2\ \ \mathbf{1}\ \ 3\ \ 2\ \ 6\ \ 5\ \ 4\ \ 3\ \ 12\ \ 1 \\
\equiv \alpha^5 &= 1\ \ 345\ \ 23\ \ \mathbf{12}\ \ 6\ \ 5\ \ 4\ \ 3\ \ 12\ \ 1 \\
&\ \ \vdots \\
\equiv \alpha^8 &= \mathbf{1}\ \ 345\ \ 23\ \ 6\ \ 5\ \ 4\ \ 123\ \ 12\ \ 1 \\
\equiv \alpha^9 &= 345\ \ \mathbf{123}\ \ 6\ \ 5\ \ 4\ \ 123\ \ 12\ \ 1 \\
\equiv \alpha^{10} &= 345\ \ 6\ \ \mathbf{123}\ \ 5\ \ 4\ \ 123\ \ 12\ \ 1 \\
\equiv \alpha^{11} &= 3456\ \ 5\ \ 1234\ \ 123\ \ 12\ \ 1 = \gamma
\end{aligned}$$

On the other hand the natural basic word that we obtain by applying $\leq_1$ on $\alpha = 654321543254364$ appears to be the natural word of the corresponding permutation.

$$
\begin{aligned}
\alpha = \alpha^0 =& 6\ 5\ 4\ 3\ 2\ \mathbf{1}\ 5\ 4\ 3\ 2\ 5\ 4\ 3\ 6\ 4\\
&\vdots\\
\equiv \alpha^3 =& 6\ 5\ 4\ 3\ 2\ 5\ 4\ 3\ \mathbf{12}\ 5\ 4\ 3\ 6\ 4\\
&\vdots\\
\equiv \alpha^5 =& 6\ 5\ 4\ 3\ 2\ 5\ 4\ 3\ 5\ 4\ \mathbf{123}\ 6\ 4\\
&\vdots\\
\equiv \alpha^7 =& 6\ 5\ 4\ 3\ \mathbf{2}\ 5\ 4\ 3\ 5\ 4\ 6\ 1234\\
&\vdots\\
\equiv \alpha^9 =& 6\ 5\ 4\ 3\ 5\ 4\ \mathbf{23}\ 5\ 4\ 6\ 1234\\
\equiv \alpha^{10} =& 6\ 5\ 4\ 3\ 5\ 45\ \mathbf{234}\ 6\ 1234\\
\equiv \alpha^{11} =& 6\ 5\ 4\ \mathbf{3}\ 5\ 456\ 234\ 1234\\
\equiv \alpha^{12} =& 6\ 5\ 45\ 3456\ 234\ 1234 = \gamma
\end{aligned}
$$

4.1.1. An algorithm on $\leq_2$.

In the following starting from a natural basic word of a permutation w we will obtain the natural word of w by applying $\leq_2$ in a particular manner:

Let $\gamma = \gamma_1 \dots \gamma_n$ be a natural basic word. Therefore its tower word decomposition is of the form

$$\gamma = b_1(b_1+1)\dots(b_1+k_1)\dots b_i(b_i+1)\dots(b_i+k_i)\dots b_r(b_r+1)\dots(b_r+k_r)$$

for some nonnegative integer $k_1,\dots,k_2$, where b_i and b_i+k_i are respectively smallest and largest integers in the i-th tower word of γ and moreover

$$b_{i+1} < (b_i+k_i) \ \text{ for all } \ 1 \leq i \leq r-1.$$

Then we have either $b_i \leq b_{i+1} < (b_i+k_i)$ or $b_{i+1} < b_i$.

In the first case it is easy to observe that the subword $b_{i+1}(b_{i+1}+1)$ is contained in the tower word $b_i(b_i+1)\dots(b_i+k_i)$. Moreover two words

$$b_i(b_i+1)\dots(b_i+k_i)\underline{b_{i+1}} \ \text{ and } \ \underline{(b_{i+1}+1)}b_i(b_i+1)\dots(b_i+k_i)$$

can be obtained from one another through a sequence of short braid relation together with the unique long braid relation applied on $b_{i+1}(b_{i+1}+1)b_{i+1}$.

Another important fact about the case $b_i \leq b_{i+1} < (b_i+k_i)$ is that the largest number of the tower word $b_{i+1}\dots(b_{i+1}+k_{i+1})$ must be strictly less then (b_i+k_i), since other wise the subword

$$b_i\dots(b_i+k_i)b_{i+1}\dots(b_{i+1}+k_{i+1})$$

of γ will not be reduced, which is clearly a contradiction. Furthermore two words

$b_i \dots (b_i+k_i)\underline{b_{i+1} \dots (b_{i+1}+k_{i+1})}$ and $\underline{(b_{i+1}+1)\dots(b_{i+1}+k_{i+1}+1)}b_i \dots (b_i+k_i)$

are related by a sequence of short and long braid relation.

Now we are ready to explain the algorithm which obtains from any natural basic word the unique natural word: So let γ be e natural basic word whose tower word decomposition is of the following form.

$$\gamma = \gamma^0 = b_1(b_1+1)\dots(b_1+k_1)\dots b_i(b_i+1)\dots(b_i+k_i)\dots b_r(b_r+1)\dots(b_r+k_r)$$

If $b_i > b_{i+1}$ for all $1 \leq i \leq r$ then γ is the unique natural word. Otherwise let j is the largest index such that $b_{j-1} \leq b_j$ and let γ^1 is the word which is obtained by first increasing every number by 1 in the tower word $b_j(b_j+1)\dots(b_j+k_j)$ and moving the resulting tower word to the left of $b_{j-1}(b_{j-1}+1)\dots(b_{j-1}+k_{j-1})$ in γ^0 i.e.,

$$\gamma^0 = \dots\dots\dots\dots b_{j-1}(b_{j-1}+1)\dots(b_{j-1}+k_{j-1})\ \ \underline{b_j(b_j+1)\dots(b_j+k_j)}\dots\dots$$
$$\gamma^1 = \dots\underline{(b_j+1)(b_j+2)\dots(b_j+k_j+1)}\ \ b_{j-1}(b_{j-1}+1)\dots(b_{j-1}+k_{j-1})\dots\dots$$

As it is discussed above γ^1 is braid equivalent to γ^0, in fact

$$\gamma^0 < \gamma^1.$$

Now if b_j+1 is smaller than the last number of the preceding tower in γ^1, namely $(b_{j-2}+k_{j-2})$, then it is a natural basic word. Otherwise, one can move all the numbers in $(b_j+1)(b_j+2)\dots(b_j+k_j+1)$ to the left of $b_{j-2}\dots(b_{j-2}+k_{j-2})$ in γ^1, and continue in the similar manner if necessary, until the resulting word is a natural basic word. In this case we still denote the resulting natural basic word by γ^1 which still satisfies $\gamma^0 < \gamma^1$.

Now continuing the above algorithm on γ^1 and so on, one can obtains a sequence of natural basic words

$$\gamma = \gamma^0 < \gamma^1 < \gamma^2 \dots$$

Observe that the above sequence terminates with a natural basic word in which the sequence of first numbers of its tower words are strictly decreasing, i.e the last word is the unique natural word.

In the following examples of natural basic words, bold characters represents the tower words which are subject to move according to the above algorithm. Observe that the last natural basic word η in both example is in fact the unique natural word of the corresponding permutation.

Example 4.2.

$$\begin{aligned}
\gamma = \gamma^0 &= 2345678\ \ 234\ \ 1234567\ \ \mathbf{56} \\
\equiv \gamma^1 &= 2345678\ \ \mathbf{67}\ \ 234\ \ 1234567 \\
\equiv \gamma^2 &= 78\ \ 2345678\ \ \mathbf{234}\ \ 1234567 \\
\equiv \gamma^3 &= 78\ \ 345\ \ 2345678\ \ 1234567 = \eta
\end{aligned}$$

Example 4.3.

$$\begin{aligned}
\gamma = \gamma^0 &= 3456\ \ 5\ \ 1234\ \ 123\ \ 12\ \ \mathbf{1}\\
\equiv \gamma^1 &= 3456\ \ 5\ \ 1234\ \ 123\ \ \mathbf{2}\ \ 12\\
\equiv \gamma^2 &= 3456\ \ 5\ \ 1234\ \ \mathbf{3}\ \ 123\ \ 12\\
\equiv \gamma^3 &= 3456\ \ 5\ \ 4\ \ 1234\ \ 123\ \ \mathbf{12}\\
\equiv \gamma^4 &= 3456\ \ 5\ \ 4\ \ 1234\ \ \mathbf{23}\ \ 123\\
\equiv \gamma^5 &= 3456\ \ 5\ \ 4\ \ 34\ \ 1234\ \ \mathbf{123}\\
\equiv \gamma^6 &= 3456\ \ \mathbf{5}\ \ 4\ \ 34\ \ 234\ \ 1234\\
\equiv \gamma^7 &= 6\ \ 3456\ \ \mathbf{4}\ \ 34\ \ 234\ \ 1234\\
\equiv \gamma^8 &= 6\ \ 5\ \ 3456\ \ \mathbf{34}\ \ 234\ \ 1234\\
\equiv \gamma^9 &= 6\ \ 5\ \ 45\ \ 3456\ \ 234\ \ 1234 = \eta
\end{aligned}$$

5. Generation algorithm: from the natural word to a reduced word

5.1. An overview

The algorithm in the previous section shows that there is a canonical route from an arbitrary reduced word to the natural word. In this section, we will try to reverse this algorithm to get a generation theorem.

The generation algorithm, as the conversion algorithm suggests, consists of two parts. In the first part, we only allow tower words to pass each other with the directed-long braid relation. This application is done by the passage operation as described in Section 5.2. We call each word obtain this way a basic word. The next step is to apply the directed-short braid relation to each of the basic words obtained in the previous step which is done by a shuffle operation as explained in Section 5.3.

5.2. Basic words of a tower diagram

To define the basic words of a tower diagram, we need some preliminary definitions.

Let a be a word of length one and $\alpha = a_1, \ldots, a_r$ be a tower word. Then we define the **tracking number** of a at the tower word α by

$$\text{track}\#(a, \alpha) := \begin{cases} a-1 & \text{if } \ a, a-1 \in \alpha, \\ a & \text{if either } \ a < a_1 - 1 \ \text{ or } \ a > a_r + 1, \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Next, for any positive integer a and for any word $\alpha = \alpha_1\alpha_2 \ldots \alpha_k$ on $\mathbb{Z}^+$ given in its tower decomposition, we define

$$a_0 := a; \quad < a, \alpha >_0 = a_0\alpha_1\alpha_2 \ldots \alpha_k.$$

Also, if $\text{track}\#(a_0, \alpha_1)$ is defined then we set

$$a_1 := \text{track}\#(a_0, \alpha_1) \ \text{ and } \ < a, \alpha >_1 = \alpha_1 a_1 \alpha_2 \ldots \alpha_k.$$

Continuing in this manner, we get words

$$< a, \alpha >_i = \alpha_1 \dots \alpha_i a_i \alpha_{i+1} \dots \alpha_k,$$

for each $1 \le i \le r \le k$ where $a_i := \text{track}\#(a_{i-1}, \alpha_i)$ is defined for all $1 \le i \le r$ but $\text{track}\#(a_r, \alpha_{r+1})$ is not defined.

Now we define the **passage words of** a **through** α as

$$\begin{aligned}\text{passwords}(a, \alpha) := &\{< a, \alpha >_i = \alpha_1 \dots \alpha_i a_i \alpha_{i+1} \dots \alpha_k \\ &\mid 0 \le i \le r \ \text{ and track}\#(a_r, \alpha_{r+1}) \text{ is not defined}\}.\end{aligned}$$

On the other hand for any word $\beta = b_1 \dots b_n$ we define **the passage words of** β **through** α in the following recursive fashion:

$$\begin{aligned}\text{passwords}(\beta, \alpha) = &\text{passwords}(b_1 \dots b_n, \alpha) \\ := &\bigcup_{\tilde{\alpha} \in \text{passwords}(b_n, \alpha)} \text{passwords}(b_1 \dots b_{n-1}, \tilde{\alpha}). \\ &\vdots \\ := &\bigcup_{\tilde{\alpha} \in \text{passwords}(b_2 \dots b_n, \alpha)} \text{passwords}(b_1, \tilde{\alpha}).\end{aligned}$$

Note that if β (or α) is an empty word then we define $\text{passwords}(\beta, \alpha)$ to be the set consisting of just α (and respectively β). Finally for any sets A and B of words, we define

$$[B, A] := \bigcup_{\alpha \in A, \beta \in B} \text{passwords}(\beta, \alpha).$$

We are now ready to define basic words of a tower diagram.

Definition 5.1. *Let $\mathcal{T} = (T_1, T_2, \dots, T_k)$ be a tower diagram decomposed into its towers and let for each $1 \le i \le k$, η_i be the natural word of the tower $\mathcal{T}_i$, possibly empty. Then the* ***set of basic words*** *for $\mathcal{T}$ is defined as follows:*

$$\text{basic}(\mathcal{T}) := [[\dots[[\{\eta_k\}, \{\eta_{k-1}\}], \{\eta_{k-2}\}], \dots, \{\eta_2\}]\{\eta_1\}].$$

We defer the example to the last section where we illustrate the algorithm. The following result follows directly from the above definition. We leave the justification to the reader.

Lemma 5.2. *Let $\mathcal{T}$ be a tower diagram. Then any basic word of the tower diagram $\mathcal{T}$ is reduced and is braid related to the natural word of $\mathcal{T}$.*

5.3. Restricted shuffle

The second step of the generation algorithm is the restricted shuffle. This operation is a restriction of the well-known shuffle operation on words. Recall that, given two words α and β, a shuffle of β over α is obtained by placing the letters of β arbitrarily between the letters of α without changing the order of

letters of β. The set of all shuffles of β over α, denoted by $\mathrm{Sh}(\alpha, \beta)$ can be obtained by first concatenating β to the right of α to obtain a new word $\alpha\,\beta$ and then moving the letters of β to the left, without changing their orders, until the word $\beta\,\alpha$ is obtained.

On the other hand, the restricted shuffle employs the same idea by adding a restriction: A letter b_i of β can pass a letter a_j of α if and only if $a_j \notin \{b_i - 1, b_i, b_i + 1\}$. With this definition, it is clear that we are referring to the short braid relation. More precise definition is as follows.

Definition 5.3. *Suppose that the letters in $\alpha = a_1 \ldots a_n$ and $\beta = b_1 b_2 \cdots b_m$ are colored by red and blue respectively. A* ***restricted shuffle*** *of α with β is a word $w = w_1 \ldots w_{n+m}$ of n red and m blue letters satisfying*

- *the restrictions of w on the red and blue letters give respectively the words α and β.*
- *if the letter b_k lies to the left a_i in w, for some $1 \le k \le m$ and for some $1 \le i \le n$, then none of $\{b_k, b_k - 1, b_k + 1\}$ lies in $a_i \ldots a_n$.*

We denote by $\mathrm{ResSh}(\alpha, \beta)$ *the set of all restricted shuffle of α with β.*

For example, let $\alpha = 13425$ and $\beta = 37$. Color the word β by boldface. Then

$$\mathrm{ResSh}(\alpha, \beta) = \{13425\mathbf{37}, 1346\mathbf{3}5\mathbf{7}, 1346\mathbf{3}\ \mathbf{7}5, 134\mathbf{3}65\mathbf{7}, 134\mathbf{3}6\mathbf{7}5\}.$$

The following result follows easily from the definition of the restriction shuffle and the definitions of the braid relations. We leave the straightforward proof to the reader.

Lemma 5.4. *Let α and β be two reduced words such that the concatenation $\alpha\beta$ is also reduced. Then*

i) *any restricted shuffle of α with β is also reduced*
ii) *restricted shuffles of α with β are pairwise distinct.*

As in the case of passage words, we can generalize the above operation to a restricted shuffle of several words by induction. Let $u_1, u_2, \ldots, u_n$ be some words on $\mathbb{Z}^+$. Then we define

$$\mathrm{ResSh}(u_1, u_2, \ldots, u_n) = \bigcup_{\alpha \in \mathrm{ResSh}(u_1, u_2, \ldots u_{n-1})} \mathrm{ResSh}(\alpha, u_n).$$

5.4. Generation Theorem

We are now ready to state the generation theorem which states that any reduced word is a restricted shuffle of a basic word. More precisely, we have the following theorem.

Theorem 5.5 (Generation Theorem)**.** *Let ω be a permutation and $\mathcal{T}$ be its tower diagram. Then*

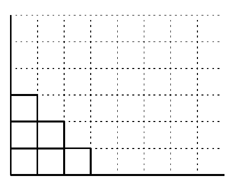
$$\mathrm{Red}(\omega) = \bigcup_{\alpha \in \mathrm{basic}(\mathcal{T})} \mathrm{ResSh}(\alpha_1, \alpha_2, \ldots, \alpha_r)$$

where $\alpha \in \mathrm{basic}(\mathcal{T})$ is given by its tower decomposition $\alpha = \alpha_1\, \alpha_2\, \ldots\, \alpha_r$.

Proof. It is clear from Lemma 5.4 and Lemma 5.2 that the right hand side is contained in the left hand side. To prove the converse inclusion, it is sufficient to show that the conversion algorithm is inverse to the generation algorithm. But clearly, any step in the conversion algorithm on $\leq_1$ is a restricted shuffle. Moreover, these steps are consistent with the order of the parenthesis in the restricted shuffle of the generation algorithm. Similarly, any natural basic word is a basic word. Indeed the reverse of each step in the conversion algorithm on $\leq_2$ is a passage word construction. $\square$

6. Example: Longest words

As our first example, we produce all reduced expressions for the longest word in S_4. The general case, for an arbitrary S_n, can be treated in the same way. Recall that the longest word is the reverse of the identity permutation and its tower diagram, $\mathcal{T}$, is given by



where the corresponding natural word η of $\mathcal{T}$, in its tower decomposition, is

$$\eta = 3\ 23\ 123.$$

Recall that the basic words of $\mathcal{T}$ is given by

$$\mathrm{basic}(\mathcal{T}) = [[\{3\}, \{23\}], \{123\}]$$

where $[\{3\}, \{23\}] = \mathrm{passwords}(\mathbf{3}, 23) = \{\mathbf{3}\,23, 23\,\mathbf{2}\}$. Therefore $\mathrm{basic}(\mathcal{T})$ is the union of $\mathrm{passwords}(\mathbf{3\,23}, 123)$ and $\mathrm{passwords}(\mathbf{23\,2}, 123)$ which are in fact the following sets:

$$\begin{aligned}
\mathrm{passwords}(\mathbf{3\,23}, 123) =& \{\mathbf{3\,23}\,123, \mathbf{3\,2}\,123\,\mathbf{2}, \mathbf{3}\,123\,\mathbf{12}, 123\,\mathbf{2\,12}, 123\,\mathbf{12\,1}\}\\
\mathrm{passwords}(\mathbf{23\,2}, 123) =& \{\mathbf{23\,2}\,123, \mathbf{23}\,123\,\mathbf{1}, \mathbf{2}\,123\,\mathbf{2\,1}, 123\,\mathbf{12\,1}\}.
\end{aligned}$$

Note that the last elements of the above sets coincide and we omit one of them. To obtain the set of all reduced words, it remains to apply restricted shuffle to

each of the basic words. We list them below.

$$\begin{aligned}
\mathrm{ResSh}(3,23,123) &= \{3\,23\,\mathbf{123}, 3\,2\,\mathbf{1}\,3\,\mathbf{23}\}\\
\mathrm{ResSh}(3,2,123,2) &= \{3\,2\,123\,2\}\\
\mathrm{ResSh}(3,123,12) &= \{3\,\mathbf{123}\,12, \mathbf{1}\,3\,\mathbf{23}\,12, 3\,\mathbf{12}\,1\,\mathbf{3}\,2, \mathbf{1}\,3\,\mathbf{2}\,1\,\mathbf{3}\,2\}\\
\mathrm{ResSh}(123,2,12) &= \{123\,2\,12\}\\
\mathrm{ResSh}(23,2,123) &= \{23\,2\,123\}\\
\mathrm{ResSh}(23,123,1) &= \{23\,\mathbf{123}\,1, 2\,\mathbf{1}\,3\,\mathbf{23}\,1, 23\,\mathbf{12}\,1\,\mathbf{3}, 2\,\mathbf{1}\,3\,\mathbf{2}\,1\,\mathbf{3}\}\\
\mathrm{ResSh}(2,123,2,1) &= \{2\,123\,2\,1\}\\
\mathrm{ResSh}(123,12,1) &= \{123\,\mathbf{12}\,1, 12\,\mathbf{1}\,3\,\mathbf{2}\,1\}
\end{aligned}$$

Finally, the union of all these restricted shuffles gives us the full set of reduced words for the longest permutation of the group S_4. Note that there are 16 reduced words in the union and the number coincides with the one that Stanley's formula [5] gives.

References

[1] N. Bergeron, S. Billey, *RC-graphs and Schubert polynomials*, Experiment Math. **2**, (1993), 257-269.

[2] O. Coşkun, M. Taşkın, *Tower Tableaux*, arXiv:1209.0938.

[3] S. Fomin, C. Greene, V. Reiner, M. Shimozono, *Balanced labellings and Schubert polynomials*, European J. Combinatorics, **18** (1997), 379-389.

[4] V. Reiner, M. Shimozono, *Plactification*, J. Alg. Comb. **4** (1995), 331-351.

[5] R. Stanley, *On the number of reduced decompositions of elements of Coxeter groups*, European J. Combinatorics, 5 (1984), 359-372.